

IN THE CLAIMS

1. (Currently amended) A computer-implemented method for communication and cooperative task completion among a plurality of distributed electronic agents, comprising the acts of:

registering a description of each active client agent's functional capabilities as corresponding registered functional capabilities, using an expandable, platform-independent, inter-agent language, wherein the inter-agent language includes:

a layer of conversational protocol defined by event types and parameter lists associated with one or more of the events; and
a content layer comprising one or more of goals, triggers and data elements associated with the events;

receiving a request for service as a base goal in the inter-agent language, in the form of an arbitrarily complex goal expression; and

dynamically interpreting the arbitrarily complex goal expression, said act of interpreting further comprising:

generating one or more sub-goals expressed in the inter-agent language;

constructing a goal satisfaction plan wherein the goal satisfaction plan includes:

a suitable delegation of sub-goal requests to best complete the requested service request-by using reasoning that includes one or more of domain-independent coordination strategies, domain-specific reasoning, and application-specific reasoning comprising rules and learning algorithms; and

dispatching each of the sub-goals to a selected client agent for performance, based on a match between the sub-goal being dispatched and the registered functional capabilities of the selected client agent.

2. (Previously presented) A computer-implemented method as recited in claim 1, further including the following acts of:

receiving a new request for service as a base goal using the inter-agent language, in the form of another arbitrarily complex goal expression, from at least one of the selected client agents in response to the sub-goal dispatched to said agent; and recursively applying the step of dynamically interpreting the arbitrarily complex goal expression in order to perform the new request for service.

3. (Previously presented) A computer-implemented method as recited in claim 2 wherein the act of registering a specific agent further includes:
invoking the specific agent in order to activate the specific agent;
instantiating an instance of the specific agent; and
transmitting the new agent profile from the specific agent to a facilitator agent in response to the instantiation of the specific agent.

4. (original) A computer-implemented method as recited in claim 1 further including the act of deactivating a specific client agent no longer available to provide services by deleting the registration of the specific client agent.

5. original) A computer-implemented method as recited in claim 1 further comprising the act of providing an agent registry data structure.

6. (original) A computer-implemented method as recited in claim 5 wherein the agent registry data structure includes at least one symbolic name for each active agent.

7. (original) A computer-implemented method of recited in claim 5 wherein the agent registry data structure includes at least one data declaration for each active agent.

8. (original) A computer-implemented method as recited in claim 5 wherein the agent registry data structure includes at least one trigger declaration for one active agent.

9. (original) A computer-implemented method as recited in claim 5 wherein the agent registry data structure includes at least one task declaration, and process characteristics for each active agent.

10. (original) A computer-implemented method as recited in claim 5 wherein the agent registry data structure includes at least one process characteristic for each active agent.

11. (original) A computer-implemented method as recited in claim 1 further comprising the act of establishing communication between the plurality of distributed agents.

12. (original) A computer-implemented method as recited in claim 1 further comprising the acts of:

receiving a request for service in a second language differing from the inter-agent language;

selecting a registered agent capable of converting the second language into the inter-agent language; and

forwarding the request for service in a second language to the registered agent capable of converting the second language into the inter-agent language, implicitly requesting that such a conversion be performed and the results returned.

13. (original) A computer-implemented method as recited in claim 12 wherein the request includes a natural language query, and the registered agent capable of converting the second language into the inter-agent language service is a natural language agent.

14. (original) A computer-implemented method as recited in claim 13 wherein the natural language query was generated by a user interface agent.

15. (original) A computer-implemented method as recited in claim 1, wherein the base goal requires setting a trigger having conditional functionality and consequential functionality.

16. (original) A computer-implemented method as recited in claim 15 wherein the trigger is an outgoing communications trigger, the computer implemented method further including the acts of:

monitoring all outgoing communication events in order to determine whether a specific outgoing communication event has occurred; and
in response to the occurrence of the specific outgoing communication event, performing the particular action defined by the trigger.

17. (original) A computer-implemented method as recited in claim 15 wherein the trigger is an incoming communications trigger, the computer implemented method further including the acts of:

monitoring all incoming communication events in order to determine whether a specific incoming communication event has occurred; and

in response to the occurrence of a specific incoming communication event satisfying the trigger conditional functionality, performing the particular consequential functionality defined by the trigger.

18. (original) A computer-implemented method as recited in claim 15 wherein the trigger is a data trigger, the computer implemented method further including the acts of:
monitoring a state of a data repository; and
in response to a particular state event satisfying the trigger conditional functionality, performing the particular consequential functionality defined by the trigger.

19. (original) A computer-implemented method as recited in claim 15 wherein the trigger is a time trigger, the computer implemented method further including the acts of:
monitoring for the occurrence of a particular time condition; and
in response to the occurrence of a particular time condition satisfying the trigger conditional functionality, performing the particular consequential functionality defined by the trigger.

20. (original) A computer-implemented method as recited in claim 15 wherein the trigger is installed and executed within the facilitator agent.

21. (original) A computer-implemented method as recited in claim 15 wherein the trigger is installed and executed within a first service-providing agent.

22. (original) A computer-implemented method as recited in claim 15 wherein the conditional functionality of the trigger is installed on a facilitator agent.

23. (original) A computer-implemented method as recited in claim 22 wherein the consequential functionality is installed on a specific service-providing agent other than a facilitator agent.

24. (original) A computer-implemented method as recited in claim 15 wherein the conditional functionality of the trigger is installed on specific service-providing agent other than a facilitator agent.

25. (original) A computer-implemented method as recited in claim 15 wherein the consequential functionality of the trigger is installed on a facilitator agent.

26. (original) A computer-implemented method as recited in claim 1 wherein the base goal is a compound goal having sub-goals separated by operators.

27. (original) A computer-implemented method as recited in claim 26 wherein the type of available operators includes a conjunction operator, a disjunction operator, and a conditional execution operator.

28. (original) A computer-implemented method as recited in claim 27 wherein the type of available operators further includes a parallel disjunction operator that indicates that disjunct goals are to be performed by different agents.

29. (Currently amended) A computer program stored on a computer readable medium, the computer program executable to facilitate cooperative task completion within a distributed computing environment, the distributed computing environment including a plurality of autonomous electronic agents, the distributed computing environment supporting an Interagent Communication Language, the computer program comprising computer executable instructions for:

providing an agent registry that declares capabilities of service-providing electronic agents currently active within the distributed computing environment;
interpreting a service request in order to determine a base goal that may be a compound, arbitrarily complex base goal, the service request adhering to an Interagent Communication Language (ICL), wherein the ICL includes:

a layer of conversational protocol defined by event types and parameter lists associated with one or more of the events; and
a content layer comprising one or more of goals, triggers and data elements associated with the events;

the act of interpreting including the sub-acts of:

determining any task completion advice provided by the base goal, and
determining any task completion constraints provided by the base goal;

constructing a base goal satisfaction plan including the sub-acts of:

determining whether the requested service is available,
determining sub-goals required in completing the base goal by using

reasoning that includes one or more of domain-independent coordination strategies, domain-specific reasoning, and application-specific reasoning comprising rules and learning algorithms,

selecting service-providing electronic agents from the agent registry suitable for performing the determined sub-goals, and

ordering a delegation of sub-goal requests to best complete the requested service; and

implementing the base goal satisfaction plan.

30. (original) A computer program as recited in claim 29 wherein the computer executable instruction for providing an agent registry includes the following computer executable instructions for registering a specific service-providing electronic agent into the agent registry:

establishing a bi-directional communications link between the specific agent and a facilitator agent controlling the agent registry;

providing a new agent profile to the facilitator agent, the new agent profile defining publicly available capabilities of the specific agent; and

registering the specific agent together with the new agent profile within the agent registry, thereby making available to the facilitator agent the capabilities of the specific agent.

31. (original) A computer program as recited in claim 30 wherein the computer executable instruction for registering a specific agent further includes:
invoking the specific agent in order to activate the specific agent;
instantiating an instance of the specific agent; and
transmitting the new agent profile from the specific agent to the facilitator agent in response to the instantiation of the specific agent.

32. (original) A computer program as recited in claim 29 wherein the computer executable instruction for providing an agent registry includes a computer executable instruction for removing a specific service-providing electronic agent from the registry upon determining that the specific agent is no longer available to provide services.

33. (original) A computer program as recited in claim 29 wherein the provided agent registry includes a symbolic name, a unique address, data declarations, trigger declarations, task declarations, and process characteristics for each active agent.

34. (original) Computer program as recited in claim 29 further including computer executable instructions for receiving the service request via a communications link established with a client.

35. (original) A computer program as recited in claim 29 wherein the computer executable instruction for providing a service request includes instructions for:
receiving a non-ICL format service request;
selecting an active agent capable of converting the non-ICL formal service request into an ICL format service request;
forwarding the non-ICL format service request to the active agent capable of converting the non-ICL format service request, together with a request that such conversion be performed; and

receiving an ICL format service request corresponding to the non-ICL format service request.

36. (original) A computer program as recited in claim 35 wherein the non-ICL format service request includes a natural language query, and the active agent capable of converting the non-ICL format service request into an ICL format service request is a natural language agent.

37. (original) A computer program as recited in claim 36 wherein the natural language query is generated by a user interface agent.

38. (original) A computer program as recited in claim 29, the computer program further including computer executable instructions for implementing a base goal that requires setting a trigger having conditional and consequential functionality.

39. (original) A computer program as recited in claim 38 wherein the trigger is an outgoing communications trigger, the computer program further including computer executable instructions for:
monitoring all outgoing communication events in order to determine whether a specific outgoing communication event has occurred; and
in response to the occurrence of the specific outgoing communication event, performing the particular action defined by the trigger.

40. (original) A computer program as recited in claim 38 wherein the trigger is an incoming communications trigger, the computer program further including computer executable instructions for:
monitoring all incoming communication events in order to determine whether a specific incoming communication event has occurred; and
in response to the occurrence of the specific incoming communication event, performing the particular action defined by the trigger.

41. (original) A computer program as recited in claim 38 wherein the trigger is a data trigger, the computer program further including computer executable instructions for: monitoring a state of a data repository; and in response to a particular state event, performing the particular action defined by the trigger.

42. (original) A computer program as recited in claim 38 wherein the trigger is a time trigger, the computer program further including computer executable instructions for: monitoring for the occurrence of a particular time condition; and in response to the occurrence of the particular time condition, performing the particular action defined by the trigger.

43. (original) A computer program as recited in claim 38 further including computer executable instructions for installing and executing the trigger within the facilitator agent.

44. (original) A computer program as recited in claim 38 further including computer executable instructions for installing and executing the trigger within a first service-providing agent.

45. (original) A computer program as recited in claim 29 further including computer executable instructions for interpreting compound goals having sub-goals separated by operators.

46. (original) A computer program as recited in claim 45 wherein the type of available operators includes a conjunction operator, a disjunction operator, and a conditional execution operator.

47. (original) A computer program as recited in claim 46 wherein the type of available operators further includes parallel disjunction operator that indicates that distinct goals are to be performed by different agents.

48. (Currently amended) An Interagent Communication Language (ICL) providing a basis for facilitated cooperative task completion within a distributed computing environment having a facilitator agent and a plurality of autonomous service-providing electronic agents, wherein:

the ICL having one or more of:

a layer of conversational protocol defined by event types and parameter lists associated with one or more of the events; and

a content layer comprising one or more of goals, triggers and data elements associated with the events;

the ICL having one or more features from a set of features comprising:

enabling agents to perform queries of other agents;

enabling agents to exchange information with other agents; and

enabling agents to set triggers within other agents; and

the ICL having a syntax supporting compound goal expressions wherein said

compound goal expressions are such that goals within a single request provided according to the ICL syntax may be coupled by one or more operators from a set of operators comprising:

a conditional execution operator; and

a parallel disjunctive operation that indicates that disjunct goals are to be performed by different agents.

49. (original) An ICL as recited in claim 48, wherein the ICL is computer platform independent.

50. (original) An ICL as recited in claim 48 wherein the ICL is independent of computer programming languages which the plurality of agents are programmed in.

51. (original) An ICL as recited in claim 48 wherein the ICL syntax supports explicit task completion constraints include use of specific agent constraints and response time constraints.

52. (original) An ICL as recited in claim 51, wherein possible types of task completion constraints include use of specific agent constraints and response time constraints.

53. (original) An ICL as recited in claim 51 wherein the ICL syntax supports explicit task completion advisory suggestions within goal expressions.

54. (original) An ICL as recited in claim 48 wherein the ICL syntax supports explicit task completion advisory suggestions within goal expressions.

55. (original) An ICL as recited in claim 48 wherein each autonomous service-providing electronic agent defines and publishes a set of capability declarations or solvables, expressed in ICL, that describes services provided by such electronic agent.

56. (original) An ICL as recited in claim 55 wherein an electronic agent's solvables define an interface for the electronic agent.

57. (original) An ICL as recited in claim 56 wherein the facilitator agent maintains an agent registry making available a plurality of electronic agent interfaces.

58. (original) An ICL as recited in claim 57 wherein the possible types of solvables includes procedure solvables, a procedure solvable operable to implement a procedure such as a test or an action.

59. (original) An ICL as recited in claim 58 wherein the possible types of solvables further includes data solvables, a data solvable operable to provide access to a collection of data.

60. (original) An ICL as recited in claim 58 wherein the possible types of solvables includes data solvables, a data solvable operable to provide access to a collection of data.

61. (Currently amended) A facilitator agent arranged to coordinate cooperative task completion within a distributed computing environment having a plurality of autonomous service-providing electronic agents, the facilitator agent comprising: an agent registry that declares capabilities of service-providing electronic agents currently active within the distributed computing environment; and a facilitating engine operable to parse a service request in order to interpret a compound goal set forth therein, the compound goal including both local and global constraints and control parameters, the service request formed according to an Interagent Communication Language (ICL), wherein the ICL includes:
a layer of conversational protocol defined by event types and parameter lists associated with one or more of the events; and
a content layer comprising one or more of goals, triggers and data elements associated with the events;
the facilitating engine further operable to construct a goal satisfaction plan by using reasoning that includes one or more of domain-independent coordination strategies, domain-specific reasoning, and application-specific reasoning comprising rules and learning algorithms.

62. (original) A facilitator agent as recited in claim 61, wherein the facilitating engine is capable of modifying the goal satisfaction plan during execution, the modifying initiated by events such as new agent declarations within the agent registry, decisions made by remote agents, and information provided to the facilitating engine by remote agents.

63. (original) A facilitator agent as recited in claim 61 wherein the agent registry includes a symbolic name, a unique address, data declarations, trigger declarations, task declarations, and process characteristics for each active agent.

64. (original) A facilitator agent as recited in claim 61 wherein the facilitating engine is operable to install a trigger mechanism requesting that a certain action be taken when a certain set of conditions are met.

65. (original) A facilitator agent as recited in claim 64 wherein the trigger mechanism is a communication trigger that monitors communication events and performs the certain action when a certain communication event occurs.

66. (original) A facilitator agent as recited in claim 64 wherein the trigger mechanism is a data trigger that monitors a state of a data repository and performs the certain action when a certain data state is obtained.

67. (original) A facilitator agent as recited in claim 66 wherein the data repository is local to the facilitator agent.

68. (original) A facilitator agent as recited in claim 66 wherein the data repository is remote from the facilitator agent.

69. (original) A facilitator agent as recited in claim 64 wherein the trigger mechanism is a task trigger having a set of conditions.

70. (original) A facilitator agent as recited in claim 61, the facilitator agent further including a global database accessible to at least one of the service-providing electronic agents.

71. (Currently amended) A software-based, flexible computer architecture for communication and cooperation among distributed electronic agents, the architecture contemplating a distributed computing system comprising:
a plurality of service-providing electronic agents; and
an Interagent Communication Language (ICL), wherein the inter-agent language includes:

a layer of conversational protocol defined by event types and parameter lists associated with one or more of the events; and
a content layer comprising one or more of goals, triggers and data elements associated with the events; and

a facilitator agent in bi-directional communications with the plurality of service-providing electronic agents, the facilitator agent including:

an agent registry that declares capabilities of service-providing electronic agents currently active within the distributed computing environment;

a facilitating engine operable to parse a service request in order to interpret an arbitrarily complex goal set forth therein, the facilitating engine further operable to construct a goal satisfaction plan including the coordination of a suitable delegation of sub-goal requests to best complete the requested service by using reasoning that includes one or more of domain-independent coordination strategies, domain-specific reasoning, and application-specific reasoning comprising rules and learning algorithms.

72. (Currently amended) A computer architecture as recited in claim 71, wherein the ~~basis for the computer architect is an~~ Interagent Communication Language (ICL) is for enabling agents to perform queries of other agents, exchange information with other agents, and set triggers within other agents, the ICL further defined by an ICL syntax supporting compound goal expressions such that goals within a single request provided according to the ICL syntax may be coupled by a conjunctive operator, a disjunctive operator, a conditional execution operator, and a parallel disjunctive operator parallel disjunctive operator that indicates that disjunct goals are to be performed by different agents.

73. (original) A computer architecture as recited in claim 72, wherein the ICL is computer platform independent.

74. (original) A computer architecture as recited in claim 73 wherein the ICL is independent of computer programming languages in which the plurality of agents are programmed.

75. (original) A computer architecture as recited in claim 73 wherein the ICL syntax supports explicit task completion constraints within goal expressions.

76. (original) A computer architecture as recited in claim 75 wherein possible types of task completion constraints include use of specific agent constraints and response time constraints.

77. (original) A computer architecture as recited in claim 75 wherein the ICL syntax supports explicit task completion advisory suggestions within goal expressions.

78. (original) A computer architecture as recited in claim 73 wherein the ICL syntax supports explicit task completion advisory suggestions within goal expressions.

79. (original) A computer architecture as recited in claim 73 wherein each autonomous service-providing electronic agent defines and publishes a set of capability declarations or solvables, expressed in ICL, that describes services provided by such electronic agent.

80. (original) A computer architecture as recited in claim 79 wherein an electronic agent's solvables define an interface for the electronic agent.

81. (original) A computer architecture as recited in claim 80 wherein the possible types of solvables includes procedure solvables, a procedure solvable operable to implement a procedure such as a test or an action.

82. (original) A computer architecture as recited in claim 81 wherein the possible types of solvables further includes data solvables, a data solvable operable to provide access to a collection of data.

83. (original) A computer architecture as recited in claim 82 wherein the possible types of solvables includes a data solvable operable to provide access to modify a collection of data.

84. (Previously presented) A computer architecture as recited in claim 71 wherein a planning component of the facilitating engine are distributed across at least two computer processes.

85. (Previously presented) A computer architecture as recited in claim 71 wherein an execution component of the facilitating engine is distributed across at least two computer processes.

86. (Currently amended) A data wave carrier providing a transport mechanism for information communication in a distributed computing environment having at least one facilitator agent and at least one active client agent, and an Interagent Communication Language (ICL), wherein the ICL includes:

a layer of conversational protocol defined by event types and parameter lists associated with one or more of the events; and
a content layer comprising one or more of goals, triggers and data elements associated with the events;

wherein said at least one facilitator agent is operable to construct a goal satisfaction plan by using reasoning that includes one or more of domain-independent coordination strategies, domain-specific reasoning, and application-specific reasoning comprising rules and learning algorithms for satisfying one or more requests for service from said at least one active client agent, the data wave carrier comprising a signal representation of an inter-agent language description of an active client agent's functional capabilities.

87. (Previously presented) A data wave carrier as recited in claim 86, the data wave carrier further comprising a corresponding signal representation of said one or more requests for service in the inter-agent language from a first agent to a second agent.

88. (Previously presented) A data wave carrier as recited in claim 86, the data wave carrier further comprising a signal representation of a goal dispatched to an agent for performance from a facilitator agent.

89. (original) A data wave carrier as recited in claim 88 wherein a later state of the data wave carrier comprises a signal representation of a response to the dispatched goal including results and/or a status report from the agent for performance to the facilitator agent.